THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 5, May–June 1981
Printed in U.S.A.

Overflow Models for *Dimension*® PBX Feature Packages

By L. KAUFMAN, J. B. SEERY, and J. A. MORRISON

(Manuscript received October 10, 1980)

We present numerical results for some traffic overflow systems with queuing. Traffic is offered by two independent streams to two groups of trunks, with a finite number of waiting spaces for each, and some overflow capability from the primary group to the secondary group. We consider three different overflow systems, two of which model feature packages offered in Dimension® PBX. *For given offered loads and unit mean holding time, we determine the number of trunks and waiting spaces in the two groups so that the blocking probabilities, and the average delays of queued calls do not exceed prescribed values. We also calculate various other quantities, such as the occupancies of the trunk groups and the probability of overflow from the primary group to the secondary group. Finally, we examine the effect on the blocking probabilities and the average delays of varying the loads offered to a given system.*

I. INTRODUCTION

In this paper we present numerical results for some traffic overflow systems with queuing. Traffic is offered by two independent streams to two groups of trunks with a finite number of waiting spaces for each, and some overflow capability from the primary group to the secondary group. The holding times of the calls are independent, and exponentially distributed. In two of the three systems considered, the overflow capability models feature packages (FP) offered in *Dimension*® PBX. The third system, which is considered for comparison, differs from the other two systems in that no overflow is permitted if there is a waiting space available in the primary queue.

Since there are a finite number of trunks and waiting spaces in each group, arriving calls may be blocked and cleared from the system. For given offered loads and unit mean holding time, we determine the

number of trunks and waiting spaces in the two groups so that the blocking probabilities and the average delays of queued calls do not exceed prescribed values. We also calculate various other quantities, such as the occupancies of the trunk groups and the probability of overflow from the primary group to the secondary group. In addition, we examine the effect on the blocking probabilities and the average delays of varying the loads offered to a given system.

The numerical results are based on different techniques developed by Kaufman[1] and Morrison.[2,3] The basic problem is to solve a large sparse system of linear equations for the steady-state probabilities of the number of calls in the two groups. Kaufman used a numerical technique involving matrix separability (block diagonalization), and in addition obtained numerical solutions by means of successive over-relaxation techniques. Kaufman has also applied her techniques to overflow systems with more than two groups. Morrison confined his attention to overflow systems with two groups, and he adopted an analytical approach which considerably reduces the dimensions of the problem.

We have been able to obtain very accurate numerical results by the methods presented in this paper. Indeed the various steady-state quantities of interest obtained by the procedures of Kaufman[1] and Morrison[2,3] agree to many significant figures. These results are considerably more accurate, and less expensive to obtain, than simulation results. They may be used as a check on the accuracy of approximate methods which have been developed for dealing with overflow problems, some of which are mentioned later in this section.

We now describe in detail the overflow systems which we consider, as depicted in Fig. 1. There are n_1 trunks and q_1 waiting spaces in the primary group, and n_2 trunks and q_2 waiting spaces in the secondary group. Traffic is offered to the two groups by two independent Poisson streams S_1 and S_2, with rates λ_1 and λ_2, respectively. The holding times of the calls are independent, and exponentially distributed with mean $1/\mu$. If all n_2 trunks in the secondary group are busy when a call from stream S_2 arrives, the call is queued if one of the q_2 waiting spaces is available, otherwise it is blocked and cleared from the system. Calls waiting in the secondary queue are placed in service on a first-in first-out basis as secondary trunks become available.

Three cases are considered for the treatment of calls offered to the primary group. The first two cases model feature packages offered in *Dimension* PBX. For these two cases, if all n_1 trunks in the primary group are busy when a call in S_1 arrives, it is placed in service in the secondary group if there is a trunk available and there are no calls waiting in the secondary queue. If no trunk is available, then the call is queued in the primary group if one of the q_1 waiting spaces is

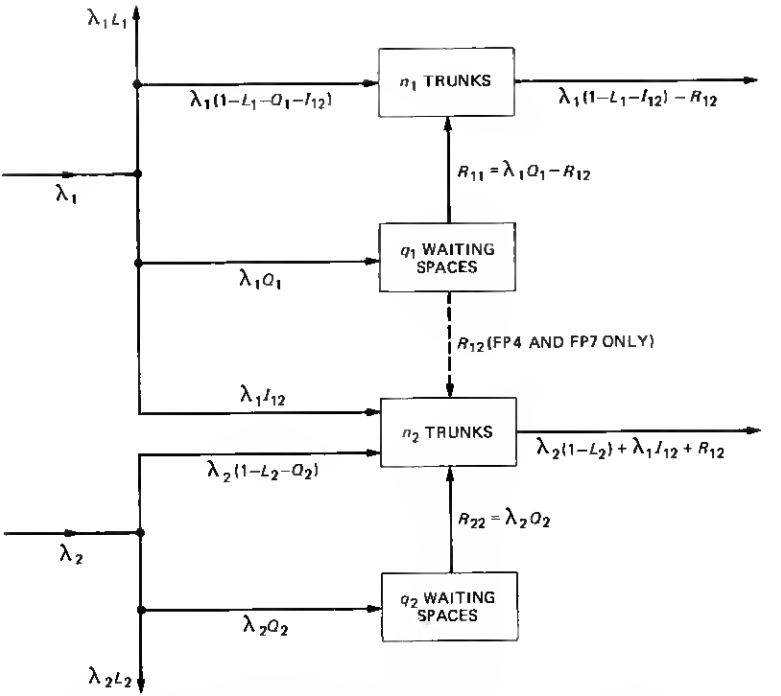

Fig. 1—Mean flow rates for an overflow system with queuing.

available, otherwise it is blocked and cleared from the system. The treatment of calls waiting in the primary queue depends on which feature package is being modeled. Corresponding to FP8, calls that are waiting in the primary queue are not allowed to overflow to the secondary group, but must wait for a trunk in the primary group to become available. Corresponding to FP4 (or FP7), calls waiting in the primary queue may be served by an idle trunk in the primary group, or by an idle trunk in the secondary group, provided that there are no calls waiting in the secondary queue, which have priority. The overflow systems corresponding to FP8 and FP4 are depicted schematically in Figs. 2 and 3, respectively.

The two cases considered above, although an idealization of the actual situation, embody the essential features of the packages. For comparison, we consider a third case, which we denote by FP0 and is depicted schematically in Fig. 4. In this case, if all n_1 trunks in the primary group are busy when a call in S_1 arrives, the call is placed in the primary queue if one of the q_1 waiting spaces is available. As with FP8, a call that is queued in the primary must wait for a trunk in the primary group to become available. If all n_1 trunks in the primary group are busy and all q_1 waiting spaces are occupied when a call in S_1

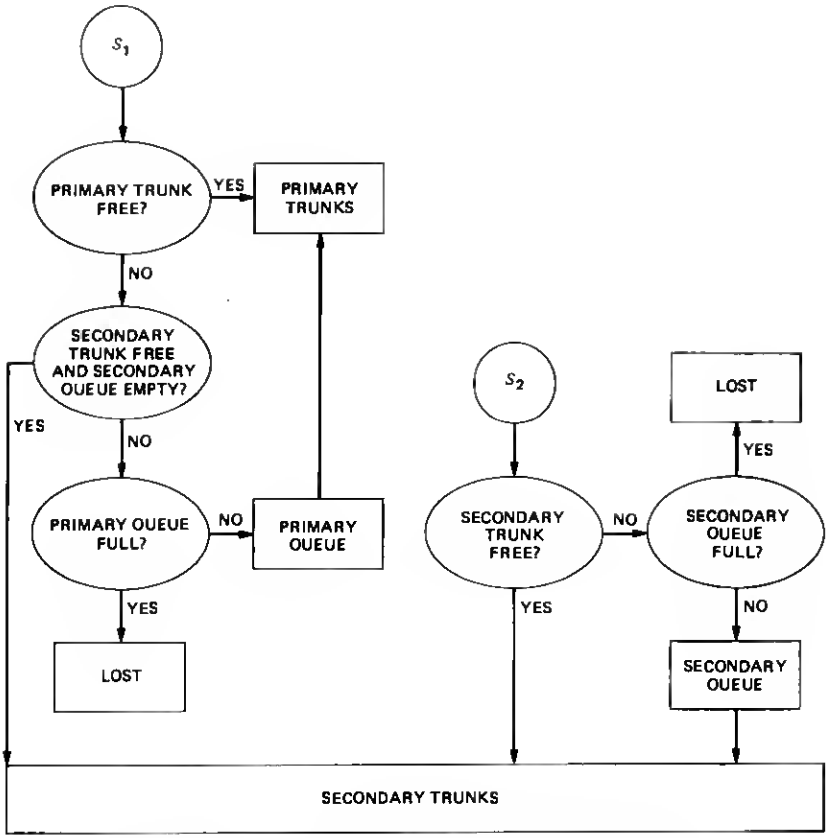

Fig. 2—Schematic of the overflow system corresponding to FP8.

arrives, it is placed in service in the secondary if there is a trunk available and there are no calls waiting in the secondary queue, otherwise it is blocked and cleared from the system. Note that no overflow is permitted if there is a waiting space available in the primary queue. This restriction was invoked by Anderson.[4]

We remark that the systems corresponding to FP4 and FP0 are particular cases of the system considered by Rath in connection with ACD-ESS (automatic call distributor-ESS).[5] He considered a system composed of two queues, in which one of the queues is allowed to overflow to the other, under specified conditions involving the queue lengths. He obtained numerical solutions in the case corresponding to FP4 by using a Gauss–Seidel iteration technique. He also developed an approximate procedure for analyzing his system based on the use of the interrupted Poisson process (IPP).[6] An approximate analysis of the system corresponding to FP4 has been given by Crater under the

assumption that the number of waiting spaces in each queue is unlimited.[7] More recently, Shulman described a method of iteration and successive approximation, using the IPP as a traffic model, for analyzing the system corresponding to FP8 with several bands of groups.[8] He has used this method in the optimal design of facilities for *Dimension* PBX with overflow and queuing features.

Section II outlines the numerical procedures for evaluating the quantitiea of interest, based on the analytical results of Morrison.[2,3] In Section III two iterative techniques used by Kaufman to obtain numerical results are discussed.[1] The numerical results are discussed in Section IV. These results were obtained by Morrison'a method and at least one of Kaufman's two methods.

II. AN ANALYTICAL PROCEDURE

We begin by outlining the numerical procedures for evaluating the quantities of interest based on the analytical results of Morrison.[2,3]

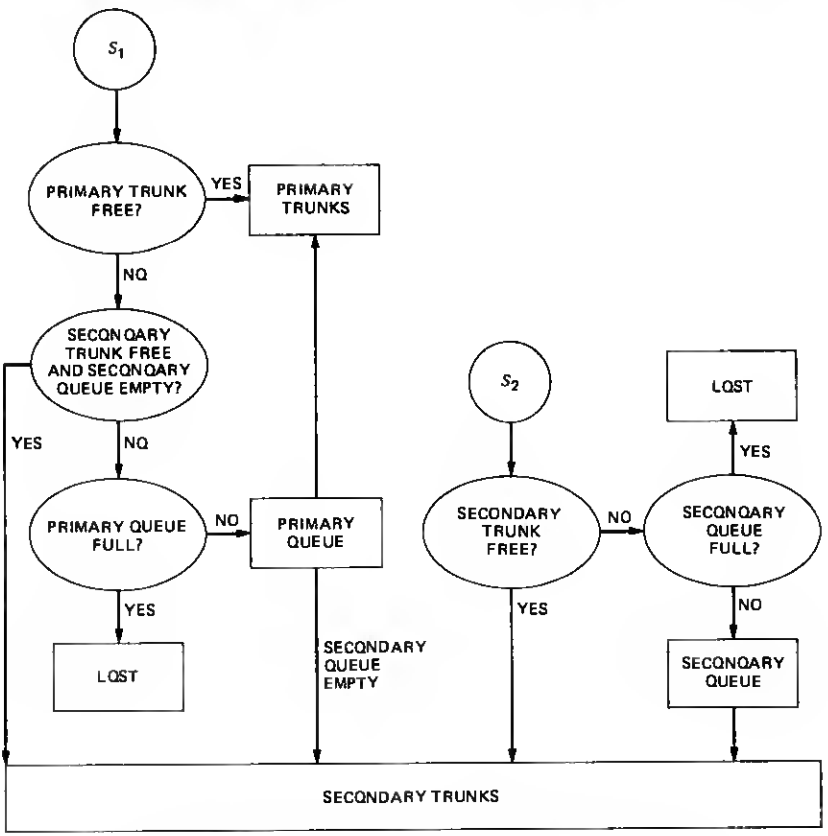

Fig. 3—Schematic of the overflow system corresponding to FP4.

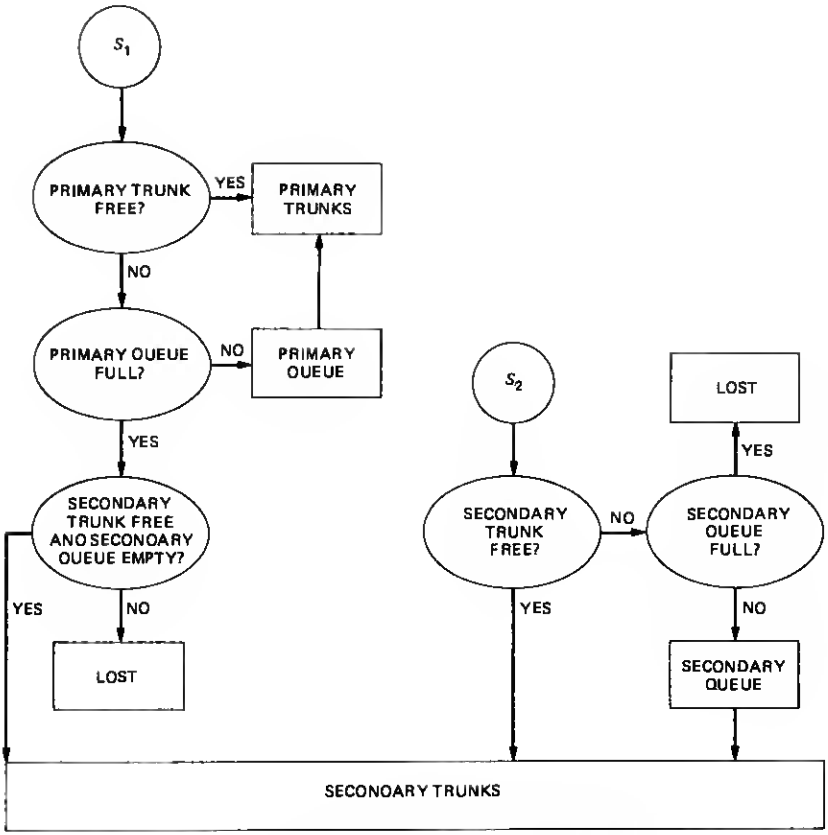

Fig. 4—Schematic of the overflow system designated as FP0.

The interested reader may consult these papers for the details of the analysis. Let p_{ij} denote the steady-state probability that there are i calls in the primary and j calls in the secondary, either in service or waiting. These probabilities satisfy a set of generalized birth-and-death equations, which take the form of partial difference equations connecting nearest-neighboring states. The basic technique is to separate variables in regions away from certain boundaries of the state space. This leads to some eigenvalue problems for the separation constant. The eigenvalues are roots of polynomial equations, and they may be evaluated numerically with the help of some interlacing properties.[2] The probabilities p_{ij} are then represented in terms of the corresponding eigenfunctions, which can be calculated from simple recurrence relations. The constant coefficients in these representations are determined from the boundary conditions (one of which is redundant), and the normalization condition that the sum of the probabilities is unity.

When engineering the system, there are various steady-state quantities of interest that can be expressed in terms of the probabilities p_{ij}. The quantities include the blocking (or loss) probabilities L_1 and L_2, and the average delays (or mean waiting times), W_1 and W_2, of calls that enter the queues. These quantities may be expressed directly in terms of the constant coefficients that occur in the representations for the probabilities p_{ij}. Thus the steady-state quantities of interest can be calculated directly, once the coefficients have been determined from the boundary and normalization conditions, without having to calculate the probabilities p_{ij}.

For FP0 there is just one set of $n_2 + q_2 + 1$ eigenvalues and corresponding eigenfunctions. There are then $n_2 + q_2 + 1$ constants to be determined numerically from the boundary and normalization conditions. We remark that in this case there are $(n_1 + q_1 + 1)(n_2 + q_2 + 1)$ probabilities p_{ij}, so that the analytical approach considerably reduces the dimensions of the problem. For FP4 there is an additional set of q_1 eigenvalues, and corresponding eigenfunctions. In this case there are $q_1 + n_2 + q_2 + 1$ constants to be determined numerically from the boundary and normalization conditions. This compares with the $q_1(q_2 + 1) + (n_1 + 1)(n_2 + q_2 + 1)$ probabilities p_{ij}. We note that there are fewer nonzero probabilities p_{ij} for FP4 since it is impossible for calls to be waiting in the primary queue when there is an idle trunk in the secondary and no calls are waiting in the secondary queue. For FP8 this is not the case, and the probabilities p_{ij} in the corresponding region of state space are expressed in terms of q_1 additional constants and a fundamental solution of a partial difference equation. However, it is possible to solve for these additional constants in terms of the other $q_1 + n_2 + q_2 + 1$ constants, by inversion of a triangular matrix.

Some conservation relations were used as a check on the accuracy of the numerical calculations. These involve the steady-state quantities depicted in Fig. 1. Thus Q_1 and Q_2 are the probabilities that calls from streams S_1 and S_2, respectively, are queued upon arrival. The mean departure rate from the primary queue to the primary trunks is R_{11}, and the mean rate of overflow from the primary queue to the secondary trunks is R_{12}. Note that $R_{12} = 0$ for both FP0 and FP8. The mean departure rate from the secondary queue, to the secondary trunks, is R_{22}. Since tbe mean rate of arrival of calls at each queue is equal to the mean departure rate, we have

$$\lambda_1 Q_1 = R_{11} + R_{12}, \qquad \lambda_2 Q_2 = R_{22}. \tag{1}$$

Also, I_{12} is the probability that a call from stream S_1 overflows immediately. Moreover, let X_1 and X_2 denote the average number of calls in service in the primary and secondary groups, respectively. According to Little's formula, the average number in a queuing system

is equal to the average arrival rate to that system times the average time spent in that system.[9] If we apply this result to the primary and secondary trunk groups, it follows that

$$\mu X_1 = \lambda_1(1 - L_1 - I_{12}) - R_{12}, \qquad \mu X_2 = \lambda_2(1 - L_2) + \lambda_1 I_{12} + R_{12}, \quad (2)$$

since $1/\mu$ is the mean holding time. Since tbe various quantities in (1) and (2) may be expressed in terms of the constant coefficients that occur in the representations for the probabilities p_{ij},[2,3] these relationships provide a useful numerical check.

A package of computer programs has been developed by Seery to calculate the various steady-state quantities of interest, based on the procedures outlined above. The documentation provides tbe information necessary to use the programs.[10]

III. SPARSE MATRIX TECHNIQUES

The birth-and-death equations which determine the steady-state probabilities, and which are mentioned in the beginning of Section II, may be written as

$$A\mathbf{p} = \mathbf{0}, \quad (3)$$

where A is a singular, nonsymmetric matrix with negative off-diagonal elements and column sums equal to zero. It has only five nonzero diagonals.

For example, for FP4 when $n_1 \geq 1$, $n_2 \geq 1$, $k_1 = n_1 + q_1$, and $k_2 = n_2 + q_2$, the matrix A is defined by the equations

$$[\lambda_1(1 - \delta_{ik_1}\chi_{j-n_2}) + \lambda_2(1 - \delta_{jk_2}) + \mu \min(i, n_1) + \mu \min(j, n_2)]p_{ij}$$

$$= (1 - \chi_{i-1-n_1}\chi_{n_2-1-j})[\lambda_1(1 - \delta_{i0})p_{i-1,j} + \mu(1 - \delta_{jk_2})\min(j + 1, n_2)p_{i,j+1}]$$

$$+ (1 - \delta_{j0})[\lambda_1\delta_{in_1}\chi_{n_2-j} + \lambda_2(1 - \chi_{i-1-n_1}\chi_{n_2-j})]p_{i,j-1}$$

$$+ \mu(1 - \delta_{ik_1})[(1 - \chi_{i-n_1}\chi_{n_2-1-j}) \min(i + 1, n_1) + n_2\chi_{i-n_1}\delta_{jn_2}]p_{i+1,j},$$

where

$$\delta_{rm} = \begin{cases} 1, & r = m \\ 0, & r \neq m \end{cases} \quad \text{and} \quad \chi_r = \begin{cases} 1, & r \geq 0 \\ 0, & r < 0 \end{cases}$$

and $0 \leq i \leq k_1$ and $0 \leq j \leq k_2$.

We will describe two iterative techniques for solving (3). The first method, based on inverse iteration, requires a few expensive iterations. The second, based on splitting A into the sum of two matrices, requires many inexpensive iterations.

Inverse iteration may be written as follows:

Pick $\mathbf{p}^{(0)}$, an approximate null vector of A.

Iterate until convergence:

For $k = 1, 2, \cdots,$

$$\text{solve } A\mathbf{v}^{(k)} = \mathbf{p}^{(k-1)} \quad \text{for} \quad \mathbf{v}^{(k)}, \tag{4}$$

$$\text{set } \mathbf{p}^{(k)} = \mathbf{v}^{(k)}/\|\mathbf{v}^{(k)}\|_1.$$

Usually only two iterations are required even if the initial guess is a random vector. Because of the near singularity of the matrix A represented in the computer, the vectors $\mathbf{v}^{(k)}$ will be very large; they will also become richer in the direction of the null space of the A matrix.[11]

Using a linear equation solver designed for band matrices, eq. (4) may be solved in approximately $3k_1k_2\max(k_1, k_2)$ locations. When $k_1k_2 > 500$, using a sparse matrix solver will require fewer operations and less space. However for larger problems (i.e., $k_1k_2 > 1000$) it pays to use some of the algebraic structure of the problem. For example, for FP4 the matrix A and the vectors $\mathbf{p}$ and $\mathbf{v}$ can be permuted and partitioned so that (4) becomes

$$\begin{pmatrix} BOX \\ OCY \\ ZQE \end{pmatrix} \begin{pmatrix} \mathbf{v}_A^{(k)} \\ \mathbf{v}_B^{(k)} \\ \mathbf{v}_C^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_A^{(k-1)} \\ \mathbf{p}_B^{(k-1)} \\ \mathbf{p}_C^{(k-1)} \end{pmatrix},$$

where $\mathbf{p}_A$ corresponds to p_{ij} where $0 \le i < n_1$ and $0 \le j \le k_2$, $\mathbf{p}_B$ corresponds to p_{ij} where $n_1 < i \le k_1$ and $n_2 < j \le k_2$, and $\mathbf{p}_C$ corresponds to p_{ij} where $i = n_1$, $0 \le j \le k_2$ and $j = n_2$, $n_1 < i \le k_1$. The matrices B and C have the form

$$B = \begin{pmatrix} S_0 & F_0 & & & \\ H & S_1 & F_1 & & \\ & \cdot & \cdot & \cdot & \\ & & H & S_{n_1-2} & F_{n_1-2} \\ & & & H & S_{n_1-1} \end{pmatrix}, \qquad C = \begin{pmatrix} G_{n_1+1} & J & & & \\ K & G_{n_1+2} & J & & \\ & \cdot & \cdot & \cdot & \\ & & K & G_{k_1-1} & J \\ & & & K & G_{k_1} \end{pmatrix},$$

where

$$F_i = -\mu(i+1)I_{k_2+1\times k_2+1}, \qquad J = -\mu n_1 I_{q_2\times q_2},$$

$$H = -\lambda_1 I_{k_2+1\times k_2+1}, \qquad K = -\lambda_1 I_{q_2\times q_2},$$

$$S_i = S + \gamma_i I_{k_2+1\times k_2+1}, \quad \text{and} \quad G_i = \gamma_i I_{q_2\times q_2} + G,$$

where

$$\gamma_i = \begin{cases} \lambda_1 + \mu \min(i, n_1), & \text{if } i < k_1, \\ \mu n_1, & \text{if } i = k_1, \end{cases}$$

and S and G have the form

$$S = \begin{pmatrix} x_0 & e_0 & & & \\ -\lambda_2 & x_1 & e_1 & & \\ & \cdot & \cdot & \cdot & \\ & & -\lambda_2 & x_{k_2-1} & e_{k_2-1} \\ & & & -\lambda_2 & x_{k_2} \end{pmatrix}, \quad G = \begin{pmatrix} x_{n_2+1} & e & & & \\ -\lambda_2 & x_{n_2+2} & e & & \\ & \cdot & \cdot & \cdot & \\ & & -\lambda_2 & x_{k_2-1} & e \\ & & & -\lambda_2 & x_{k_2} \end{pmatrix},$$

where

$$x_j = \begin{cases} \lambda_2 + \mu \min(j, n_2), & \text{if } j < k_2, \\ \mu n_2, & \text{if } j = k_2, \end{cases}$$

$$e_j = -\mu \min(j + 1, n_2),$$

$$e = -\mu n_2.$$

The matrices X, Y, Z, Q, and E are very sparse (there are only $k_2 + 1$ nonzero elements in X) but they do not possess any relevant algebraic structure as B and C do. The eigendecomposition of B can be expressed in terms of the eigendecomposition of an $n_1 \times n_1$ tridiagonal matrix and the eigendecomposition of the $(k_2 + 1) \times (k_2 + 1)$ tridiagonal matrix S. Similarly the eigendecomposition of C can be expressed in terms of the eigendecomposition of a $q_1 \times q_1$ tridiagonal matrix and the $q_2 \times q_2$ tridiagonal matrix G. Using block Gaussian elimination and the eigendecompositions of B and C, solving (4) entails formulating and solving a dense system of $k_2 + 1 + q_1$ equations.

The second method, line successive overrelaxation (SOR), is much easier to implement and can be easily generalized to more queues. it uses the fact that A and $\mathbf{p}$ can be permuted and partitioned so that (3) can be written as

$$\begin{pmatrix} T_0 & E_0 & & & \\ D_1 & T_1 & E_1 & & \\ & \cdot & \cdot & \cdot & \\ & & D_{k_1-1} & T_{k_1-1} & E_{k_1-1} \\ & & & D_{k_1} & T_{k_1} \end{pmatrix} \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \\ \\ \mathbf{p}_{k_1} \end{pmatrix} = 0,$$

where $\mathbf{p}_i$ corresponds to p_{ij}, $0 \le j \le k_2$. The T matrices are tridiagonal and the E and D matrices are diagonal. The exact elements of the matrices depend on the feature package modeled.

In line SOR, initial vectors $\mathbf{p}_i^{(0)}$ and a parameter ω are chosen and the following iteration rule is executed:

For $k = 1, 2, \cdots$ until convergence,

  for $i = 0, 1, 2, \cdots, k_1$,

    solve $T_i \mathbf{v} = D_i \mathbf{p}_{i-1}^{(k+1)} + E_i \mathbf{p}_{i+1}^{(k)}$ for $\mathbf{v}$,

    set $\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \omega(-\mathbf{v} - \mathbf{p}_i^{(k)})$.

After convergence,
set $\mathbf{p} \leftarrow \mathbf{p}/|\mathbf{p}|_1$.

The off-diagonal elements are never stored, but generated each time. Each iteration requires about $5m$ multiplications, $7m$ additions, and $2m$ divisions, where m is the size of the state space. The number of iterations depends on the ratios of the λ's to μ's, the size of the problem, the choice of ω, the feature package, and the required accuracy. Most of the examples given in Section IV required between 50 and 150 iterations for problems of between 500 and 2500 unknowns.

When $\omega = 1$, line SOR is called block Gauss–Seidel and is known to converge (see Kaufman[1]). For our problems a 10 percent change in ω sometimes doubles the number of iterations, as Table I illustrates. Usually the optimal ω is about 1.7 and choosing ω above 1.9 causes divergence. A heuristic algorithm was developed for adjusting ω as the iterative scheme progresses. The algorithm uses the theory developed in Ref. 12 to obtain an overestimate of ω and takes into consideration the fact that as ω is increased, the relative error tends to initially increase before decreasing if the choice of ω yields a convergent scheme.

Various other splitting schemes have been tried. Chebychev acceleration with Gauss–Seidel preconditioning requires about the same amount of computational effort as line SOR with the optimal ω.[12] Chebychev acceleration also requires the estimation of certain unknown parameters but the course of the algorithm seems to be much less sensitive to their values. Chebychev acceleration with an incomplete LU preconditioning has been successful for problems, such as the ones we have, in which the graph underlying the matrix is two-cyclic. A block SOR method in which each diagonal block is a separable matrix has also been tried. For the problems we have considered, the increase in the amount of work per iteration is not compensated by the decrease in the number of iterations.

Table I—ω vs number of iterations for FPO, $\mu = 1$, $\lambda_1 = 40$, $\lambda_2 = 36$, $n_1 = 40$, $n_2 = 40$, $q_1 = 10$, $q_2 = 10$, stopping criteria $\sim 10^{-9}$

ω	No. of Iterations
1.60	184
1.70	119
1.75	94
1.80	114
1.90	217

IV. NUMERICAL RESULTS

The criteria we use for engineering the system are that the blocking probabilities, L_1 and L_2, and the average delays, W_1 and W_2, of calls which enter the queues, do not exceed specified values. For prescribed loads $a_1 = \lambda_1/\mu$ and $a_2 = \lambda_2/\mu$, the aim is to choose the number of primary and secondary trunks n_1 and n_2, and the number of waiting spaces q_1 and q_2, so that the criteria are met. The procedure used was to select values of n_1 and n_2, and then determine the smallest values of q_1 and q_2 for which the criteria on the blocking probabilities are met. The process was repeated for different sets of values of n_1 and n_2, to find sets for which the criteria on the average delays are also satisfied. A search was made to determine the smallest total number of trunks $n_1 + n_2$ for which the criteria on both the blocking probabilities and the average delays are met, with appropriate choices of q_1 and q_2.

In the numerical results given below, we take $\mu = 1$, so that the average delays are given in units of mean holding time. Two sets of results were obtained. The first set corresponds to primary and secondary loads $a_1 = 3$ and $a_2 = 8$, and the desired criteria were $\max(L_1, L_2) \leq 0.01$ and $\max(W_1, W_2) \leq 1$. The results in Table II, which correspond to four primary and nine secondary trunks, indicate how changes in the number of waiting spaces affect the blocking probabilities and the average delays. The results of the search to minimize the total number of trunks required are depicted in Table III. In none of the cases were we able to satisfy the criteria with a total of only 12 trunks. Moreover, for both FP8 and FP0, for a total of 13 trunks we required 4 in the primary group and 9 in the secondary group. This was not the case for FP4. However, if we take into account the fact that primary trunks are less expensive than secondary ones, then for FP4 we would choose the set with four in the primary group

Table II—Blocking probabilities and average delays for offered loads $a_1 = 3$ and $a_2 = 8$, and unit mean holding time, $n_1 = 4$ and $n_2 = 9$ trunks, and different values of q_1 and q_2

FP	q_1	q_2	$10^2 L_1$	$10^2 L_2$	W_1	W_2
8	6	18	1.063	1.027	0.612	0.727
8	7	18	0.725	1.027	0.650	0.727
8	6	19	1.076	0.904	0.613	0.748
8	7	19	0.735	0.905	0.651	0.748
0	7	17	1.226	1.070	0.730	0.705
0	8	17	0.907	1.069	0.777	0.705
0	7	18	1.232	0.943	0.730	0.727
0	8	18	0.911	0.941	0.777	0.727
4	6	18	0.945	1.059	0.519	0.727
4	7	18	0.637	1.060	0.549	0.727
4	6	19	0.959	0.932	0.521	0.748
4	7	19	0.648	0.934	0.552	0.748

and nine in the secondary group. Also listed in Table III are other steady-state quantities of interest, as defined in Section II. In addition, $O_{12} = \lambda_1 I_{12} + R_{12}$ is the total mean overflow rate from the primary group to the secondary group.

It is of interest to compare the results for the three cases in Table III corresponding to $n_1 = 4$ and $n_2 = 9$, bearing in mind the differences in the number of waiting spaces and the results in Table II. Calls arriving at the primary group are most likely to be queued, and (by far) least likely to overflow (immediately), for FP0, as is to be expected, since overflow is permitted only when the queue is full. Moreover, the average delay in the primary queue is largest for FP0, and larger for FP8 than for FP4, since for FP8 no overflow is permitted from the primary queue. Also, the total mean overflow rate for FP4 exceeds that for FP8, although the immediate overflow probability is smaller for FP4. It is apparently the capacity for overflow from the primary queue which accounts for the other solutions for FP4 with a total of 13 trunks.

The second set of results corresponds to primary and secondary loads $a_1 = 10$ and $a_2 = 5$, and the desired criteria were $\max(L_1, L_2) \leq 0.005$ and, at first, $\max(W_1, W_2) \leq 0.8$. The results of the search with a total of 17 trunks are depicted in Table IV. There are no solutions for FP0 which satisfy the criteria, and just two for FP8. Three solutions are given for FP4. Although it is not possible to satisfy the criteria with a total of 17 trunks and more than 10 in the primary group, it is expected that there are solutions with less than 8 in the primary group, but we have not checked this. There are no solutions for FP4 or FP8 with a total of 16 trunks which satisfy the criteria. If we relax the

Table III—Steady-state quantities for offered loads $a_1 = 3$ and $a_2 = 8$, and unit mean holding time, with $\max(L_1, L_2) \leq 0.01$ and $\max(W_1, W_2) \leq 1$

FP	8	0	4	4	4	4
n_1	4	4	4	3	2	1
n_2	9	9	9	10	11	12
q_1	7	8	6	8	10	11
q_2	19	18	19	12	9	7
$10^2 L_1$	0.735	0.911	0.959	0.955	0.890	0.890
$10^2 L_2$	0.905	0.941	0.932	0.831	0.810	0.934
W_1	0.651	0.777	0.521	0.679	0.805	0.861
W_2	0.748	0.727	0.748	0.411	0.284	0.214
Q_1	0.293	0.477	0.288	0.351	0.391	0.413
Q_2	0.682	0.621	0.703	0.563	0.492	0.451
I_{12}	0.080	0.004	0.056	0.142	0.258	0.400
R_{11}	0.878	1.430	0.743	0.716	0.545	0.291
R_{12}	0	0	0.123	0.336	0.628	0.950
R_{22}	5.455	4.967	5.621	4.506	3.938	3.607
O_{12}	0.241	0.012	0.289	0.762	1.401	2.151
X_1	2.737	2.960	2.682	2.209	1.572	0.823
X_2	8.169	7.937	8.215	8.695	9.336	10.076

Table IV—Steady-state quantities for offered loads $a_1 = 10$ and $a_2 = 5$, and unit mean holding time, with $\max(L_1, L_2) \leq 0.005$ and $\max(W_1, W_2) \leq 0.8$

FP	8	8	4	4	4
n_1	10	9	10	9	8
n_2	7	8	7	8	9
q_1	20	22	19	20	21
q_2	11	8	11	9	7
$10^2 L_1$	0.481	0.473	0.473	0.480	0.445
$10^2 L_2$	0.453	0.481	0.494	0.345	0.430
W_1	0.637	0.773	0.508	0.532	0.547
W_2	0.460	0.309	0.460	0.317	0.237
Q_1	0.433	0.421	0.464	0.474	0.481
Q_2	0.626	0.538	0.683	0.623	0.583
I_{12}	0.117	0.187	0.054	0.087	0.123
R_{11}	4.327	4.208	3.921	3.622	3.268
R_{12}	0	0	0.716	1.120	1.540
R_{22}	3.131	2.692	3.417	3.114	2.915
O_{12}	1.169	1.875	1.257	1.987	2.774
X_1	8.783	8.078	8.696	7.965	7.182
X_2	6.147	6.851	6.232	6.970	7.752

criteria on the average delays to $\max(W_1, W_2) \leq 1$, then additional solutions are obtained, as depicted in Table V. There is now one solution for FP0 with a total of 17 trunks, and two more solutions for FP8. Two additional solutions are given for FP4, one with a total of only 16 trunks (and 34 waiting spaces for the primary group). There are no solutions for FP8 or FP0 with a total of 16 trunks which satisfy the relaxed criteria.

Finally, setting aside the problem of engineering the system, we examined the effect on the blocking probabilities and the average

Table V—Steady-state quantities for offered loads $a_1 = 10$ and $a_2 = 5$, and unit mean holding time, with $\max(L_1, L_2) \leq 0.005$ and $0.8 < \max(W_1, W_2) \leq 1$

FP	8	8	0	4	4
n_1	11	8	11	11	10
n_2	6	9	6	6	6
q_1	18	24	24	18	34
q_2	18	7	17	19	19
$10^2 L_1$	0.461	0.482	0.499	0.437	0.466
$10^2 L_2$	0.495	0.354	0.456	0.428	0.471
W_1	0.523	0.954	0.753	0.478	0.983
W_2	0.883	0.237	0.866	0.898	0.898
Q_1	0.437	0.406	0.654	0.452	0.676
Q_2	0.761	0.479	0.579	0.795	0.874
I_{12}	0.054	0.263	0.002	0.025	0.020
R_{11}	4.373	4.060	6.540	4.187	6.191
R_{12}	0	0	0	0.335	0.568
R_{22}	3.807	2.397	2.895	3.973	4.371
O_{12}	0.540	2.633	0.017	0.587	0.771
X_1	9.414	7.319	9.933	9.370	9.183
X_2	5.515	7.615	4.994	5.565	5.747

Table VI—Blocking probabilities and average delays for offered loads a_1 and $a_2 = 0.9a_1$, unit mean holding time, $n_1 = 40 = n_2$ trunks, $q_1 = 10 = q_2$ waiting spaces, and different values of a_1

FP	a_1	$10^2 L_1$	$10^2 L_2$	W_1	W_2
0	42	5.355	3.654	0.1475	0.1259
0	40	2.906	2.022	0.1375	0.1162
0	38	1.291	0.954	0.1270	0.1065
0	36	0.449	0.378	0.1162	0.0970
8	42	3.796	4.774	0.1313	0.1259
8	40	1.758	2.845	0.1184	0.1162
8	38	0.639	1.436	0.1055	0.1065
8	36	0.177	0.596	0.0930	0.0970
4	42	3.588	5.020	0.1179	0.1259
4	40	1.642	2.995	0.1043	0.1162
4	38	0.590	1.510	0.0909	0.1065
4	36	0.161	0.623	0.0782	0.0970

delays of varying the loads offered to a prescribed system. We did this since in a given situation the configuration of the system is fixed, but one would expect that the loads would vary in the course of the day. We chose a configuration with 40 trunks and 10 waiting spaces in both the primary and the secondary groups, to show that our programs can handle larger problems than those listed so far. The primary and secondary loads were varied simultaneously with $a_2 = 0.9a_1$, and tbe results are given in Table VI. As expected, the blocking probabilities and the average delays increase with increasing load, the more significant effect being on the blocking probabilities.

For a given load, the blocking probability for calls from stream S_1 is larger for FP0 than for FP8, and slightly larger for FP8 than for FP4. On the contrary, the blocking probability for calls from stream S_2 is smaller for FP0 than for FP8, and slightly smaller for FP8 than for FP4. These orderings are not surprising in view of the fact that no overflow is permitted from the primary queue for either FP0 or FP8, and moreover overflow is permitted for FP0 only when the primary queue is full. As before, the average delay in tbe primary queue is largest for FP0, and larger for FP8 than for FP4. The average delay in the secondary queue is, of course, the same for FP0, FP4, and FP8.

V. ACKNOWLEDGMENTS

The authors are grateful to G. M. Anderson, J. F. Brown, and T. V. Crater for bringing this problem to their attention, and to J. McKenna for his continued encouragement. They are indebted to J. H. Rath for running his program to check their initial calculations for FP4 and to H. B. Shulman for suggesting the two examples for the engineering problem, and for providing nearly optimal results for FP8.

REFERENCES

1. L. Kaufman, "Solving Large Linear Systems Arising in Queuing Problems," to appear in the proceedings of the Bielefeld conference on large linear systems, Springer Verlag.
2. J. A. Morrison, "Analysis of Some Overflow Problems With Queuing," B.S.T.J., *59*, No. 8 (October 1980), pp. 1427–62.
3. J. A. Morrison, "An Overflow System in Which Queuing Takes Precedence," B.S.T.J., *60*, No. 1 (January 1981), pp. 1–12.
4. G. M. Anderson, "Facilities Design for Automatic Route Selection With Queuing," unpublished work.
5. J. H. Rath, "An Approximation for a Queueing System With Two Queues and Overflows," unpublished work.
6. A. Kuczura, "The Interrupted Poisson Process as an Overflow Process," B.S.T.J., *52*, No. 3 (March 1973), pp. 437–48.
7. T. V. Crater, "Calculation of Trunk Occupancy and Delay for Networks With Automatic Alternate Routing and Queueing," unpublished work.
8. H. B. Shulman, "Engineering Trunk Networks for Queueing and Overflow," unpublished work.
9. L. Kleinrock, *Queueing Systems, Volume I: Theory*, New York: Wiley, 1975.
10. J. B. Seery, "FP048–A Computer Package for Overflow Problems Motivated by *Dimension*® PBX Feature Packages," unpublished work.
11. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, London: Oxford University Press, 1965.
12. R. S. Varga, *Matrix Iterative Analysis*, Englewood Cliffs, New Jersey: Prentice-Hall, 1962.